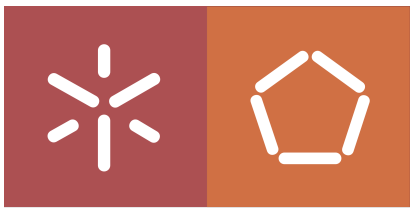

Universidade do Minho
Escola de Engenharia

Redes de Computadores

Relatório do Trabalho Prático nº3
Grupo 41

LEI - 2º Ano - 2º Semestre

Realizador por:

A98695	Lucas Oliveira
A89292	Mike Pinto
A96208	Rafael Gomes

Braga,
5 de maio de 2023

Conteúdo

Lista de Figuras

Capítulo 1

Captura e análise de Tramas Ethernet

Ative o Wireshark na sua máquina nativa. No seu browser, aceda ao URL https://alunos.uminho.pt .
Pare a captura do Wireshark., e proceda da seguinte forma:
Localize o estabelecimento da conexão entre o cliente e o servidor HTTP (sequência de tramas com as TCP flags TCP SYN, SYNACK, ACK ativas). Após a fase de estabelecimento seguro da conexão, obtenha o número de ordem da sequência de bytes capturada (coluna da esquerda na janela do Wireshark) correspondente à trama que transporta os primeiros dados aplicacionais enviados do cliente para o servidor (Application Data). Identifique também o número de ordem da trama com a resposta proveniente do servidor que contém os dados correspondentes ao acesso web realizado pelo cliente (browser). Note que os dados aplicacionais são enviados de forma segura usando o protocolo TLS (Transport Layer Security), mapeados para um segmento TCP, transportado num datagrama IP que, por sua vez, é encapsulado no campo de dados da trama Ethernet. Expanda a informação do nível da ligação de dados e observe o conteúdo da trama Ethernet (cabeçalho e dados (payload)). Responda às perguntas seguintes com base no conteúdo da trama Ethernet que contém a mensagem de acesso ao servidor (HTTP GET encriptada).

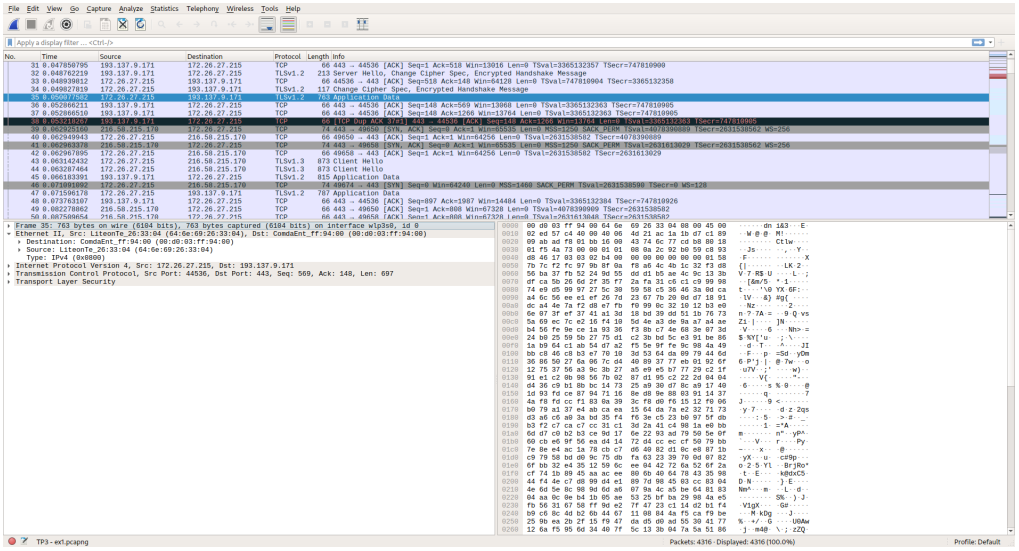

Figura 1.0.1: Captura tráfego *wireshark*.

1.1 Pergunta 1

Anote os endereços MAC de origem e de destino da trama capturada. Identifique a que sistemas se referem. Justifique.

Após análise da figura 1.0.1, os endereços *MAC* de origem e destino da trama capturada são:

- Origem: 64:6e:69:26:33:04, que corresponde à interface do *router* da rede local a que estamos conetados.
- Destino: 00:d0:03:ff:94:00, que corresponde à interface ativa da placa de rede do nosso dispositivo.

1.2 Pergunta 2

Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?

O valor hexadecimal apresentado no campo *Type* da trama *Ethernet* é *0x0800* e tem como significado o protocolo nível de rede *Ipv4*, ver figura 1.0.1.

1.3 Pergunta 3

Quantos bytes são usados no encapsulamento protocolar, i.e. desde o início da trama até ao início dos dados do nível aplicacional (Application Data Protocol: http-over-tls, no caso de HTTPS)? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar.

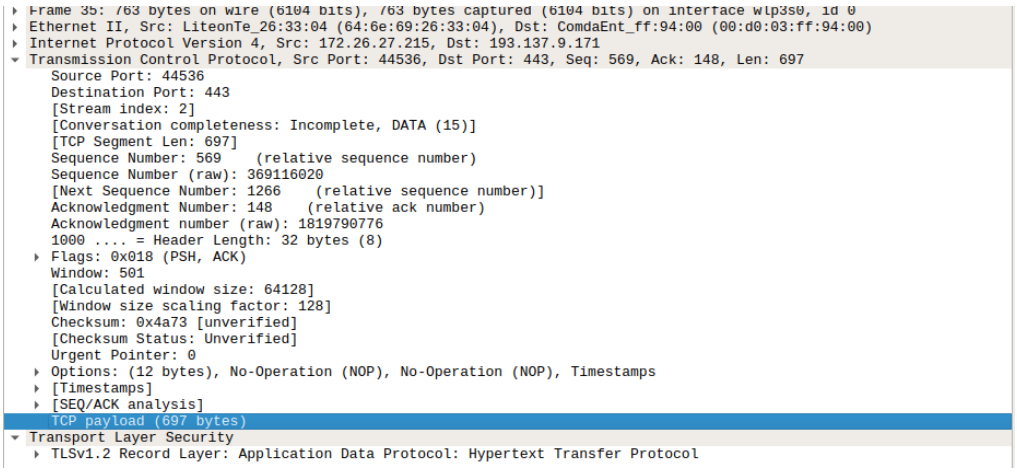

Figura 1.3.1: Campo de dados da trama capturada.

A trama capturada possui o tamanho total de 763 *bytes*. Sabemos que o tamanho de dados do protocolo *TLS* tem o tamanho de 692 *bytes* de dados e 5 bytes de *header*, como podemos verificar na figura 1.3.1. Subtraindo os valores obtemos 71 *bytes* usados no encapsulamento protocolar. A sobrecarga, em percentagem, introduzida pela pilha protocolar é aproximadamente $\frac{71}{763} = 0.09 = 9\%$.

A seguir responda às seguintes perguntas, baseado no conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP proveniente do servidor.

1.4 Pergunta 4

Qual é o endereço Ethernet da fonte? A que sistema de rede corresponde? Justifique.

O respetivo endereço *Ethernet* de fonte é 00:d0:03:ff:94:00 que corresponde à interface do *router* que estamos conectados.

1.5 Pergunta 5

Qual é o endereço MAC do destino? A que sistema (host) corresponde?

O endereço *MAC* de destino é 64:6e:69:26:33:04 que corresponde à interface ativa da placa de rede do nosso dispositivo.

1.6 Pergunta 6

Atendendo ao conceito de encapsulamento protocolar, identifique os vários protocolos contidos na trama recebida. Justifique, indicando em que campos dos cabeçalhos capturados se baseou.

Os vários protocolos contidos na trama recebida são:

- *Ethernet II*

- *IPv4* (*Internet Protocol Version 4*
- *TCP* (*Transmission Control Protocol*
- *TLS* (*Transport Layer Security)*
- *Application Data Protocol: Hypertext Transfer Protocol*

Baseamo-nos nos campos de dados da figura 1.3.1 e no campo *TLSv1.2 Record Layer*.

Capítulo 2

Protocolo ARP

Nesta secção, pretende-se analisar a operação do protocolo ARP. Nesta secção, pretende-se analisar a operação do protocolo ARP. Para tal, inicie o emulador CORE com o comando "sudo core" e mantenha-o ativo até ao final do trabalho. Crie uma topologia de rede com dois departamentos, A e B. O departamento A usará os endereços 192.168.0+turnogrupo .X/25, e o departamento B 192.168.128+turnogrupo .X/25, sendo X o decimal atribuído automaticamente pelo CORE. Por exemplo, o grupo PL22 usará os endereços 192.168.22.X/25 e 192.168.150.X/25. Adotando a terminologia usada no CORE, considere que o departamento A contém três PCs e um host (servidor) ligados a um switch, que por sua vez liga ao router RA. O departamento B tem três PCs ligados a um hub, que por sua vez liga ao router RB. Os dois routers estão ligados entre si por uma ligação física, cujo endereço de rede é atribuído automaticamente pelo CORE. Todos os links têm uma largura de banda de 200 Mbps. Para facilitar a configuração dos endereços de rede, comece por ligar apenas o switch e o hub aos routers e depois configure os endereços IP das interfaces do router de acordo com a regra definida. Seguidamente ligue os PCs e o servidor ao switch e ao hub, ficando assim automaticamente configurados com os endereços IP desejados. No sentido de observar o envio e receção de mensagens ARP, é conveniente apagar o conteúdo da cache ARP. Caso contrário, é provável que a associação entre endereços IP e MAC já exista em cache. Apague a cache ARP usando o comando arp –d. Um método expedito no CORE de apagar todas as caches ARP é reiniciar a rede. Selecione um PC de um dos departamentos à sua escolha e inicie a captura de tráfego com o Wireshark do CORE. A partir desse sistema efetue ping para dois PCs localizados na outra rede (departamento). Pare a captura de tráfego no Wireshark e localize o tráfego ARP, usando o filtro arp.

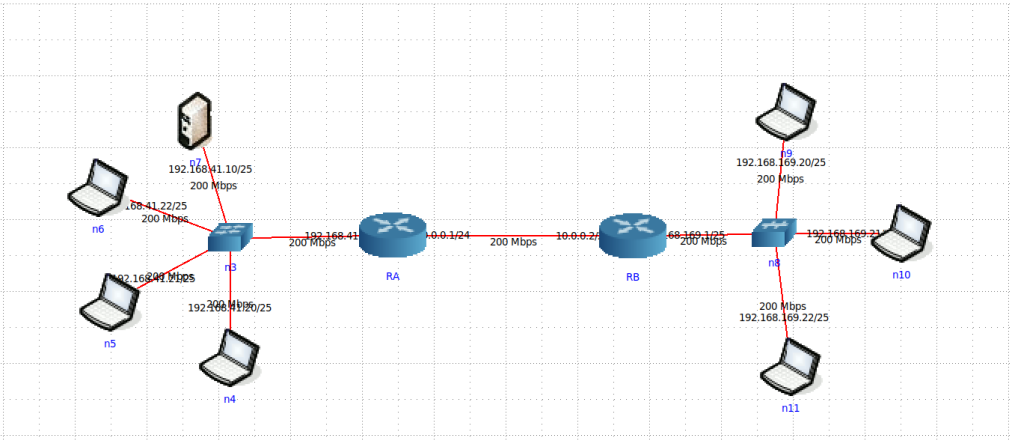

Figura 2.0.1: Topologia Core pedida.

2.1 Pergunta 1

Abra uma consola no PC onde efetuou o ping. Observe o conteúdo da tabela ARP com o comando arp -a

```
root@n5:/tmp/pycore.34293/n5.conf# arp -a
? (192.168.41.1) at 00:00:00:aa:00:00 [ether] on eth0
root@n5:/tmp/pycore.34293/n5.conf#
```

Figura 2.1.1: Tabela *ARP* dispositivo *n5*

2.1.1 Alínea a)

Com a ajuda do manual ARP (man arp), interprete o significado de cada uma das colunas da tabela.

Ao executar o comando *arp -a* no dispositivo *n5* do departamento A, obtemos a tabela *ARP* da figura 2.1.1.

A função da tabela *ARP* é mapear o endereço *IP* para endereços *MAC*. Na primeira coluna observamos um endereço *IP* do *router* R_A e na segunda coluna o respetivo endereço *MAC*.

2.1.2 Alínea b)

Indique, justificando, qual o equipamento da intranet em causa que poderá apresentar a maior tabela ARP em termos de número de entradas.

Nesta topologia de rede só existem dois equipamentos *intranet*, um *switch* conectado ao *router RA* e um *hub* conectado ao *router RB*.

Sendo que neste caso, o equipamento que poderá apresentar a maior tabela *ARP* será o *switch* pois, como os *hubs* são dispositivos pertencentes ao nível 1 da camada protocolar, não funcionam a nível 2, logo não conseguem fazer a diferenciação entre endereços *MAC*, não possuindo tabelas *ARP*.

2.2 Pergunta 2

Observe a trama Ethernet que contém a mensagem com o pedido ARP (ARP Request).

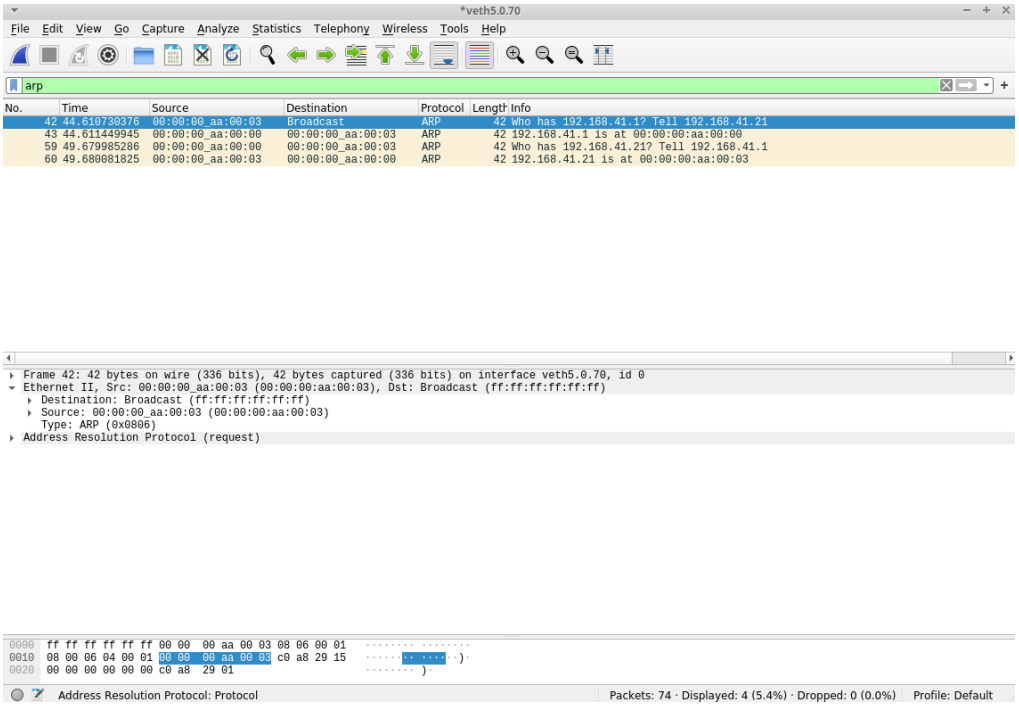

Figura 2.2.1: Captura *Wireshark* de pacotes *ARP*
RP

2.2.1 Alínea a)

Qual é o valor hexadecimal dos endereços MAC origem e destino? Como interpreta e justifica o endereço destino usado?

Pela figura 2.2.1 observamos que o endereço *MAC* de origem é 00:00:00:aa:00:03 que representa o computador n5 do departamento A, e o endereço *MAC* de destino é ff:ff:ff:ff:ff:ff que representa um endereço de *broadcast*, enviando o *ARP reply* para todos os dispositivos da rede local. Este endereço foi utilizado, pois a tabela *ARP* do dispositivo n5 antes do *ping* encontrava-se vazia, então, este realizou um *ARP Request* em *broadcast*, com o endereço MAC de destino ff:ff:ff:ff:ff:ff e endereço IP de destino o do *router* R_A com o objetivo de obter uma resposta(*ARP Reply*) do *router* R_A com o seu endereço *MAC*.

2.2.2 Alínea b)

Qual o valor hexadecimal do campo Tipo da trama Ethernet? O que indica?

Pela figura 2.2.1 podemos ver que o valor hexadecimal no tipo é *0x0806* que indica que se trata de um tipo *ARP*.

2.2.3 Alínea c)

Observando a mensagem ARP, como pode saber que se trata efetivamente de um pedido ARP? Refira duas formas distintas de obter essa informação.

Pelo campo *info* da captura do *Wireshark* da figura 2.2.1, retiramos a mensagem *"Who has 192.168.41.1 tell 192.168.41.21"* que é uma forma de identificar que se trata de uma mensagem ARP, assim como os endereços da *source* e *destination* endereços em *hexadecimal MAC*.

2.2.4 Alínea d)

Explicite, em linguagem comum, que tipo de pedido ou pergunta é feita pelo host de origem à rede?

O tipo de pedido feita pelo host de origem à rede é "*Who has X? Tell Y*". Basicamente, o dispositivo Y, pede ao dispositivo X o seu endereço *MAC*.

2.3 Pergunta 3

Localize a mensagem ARP que é a resposta ao pedido ARP efetuado

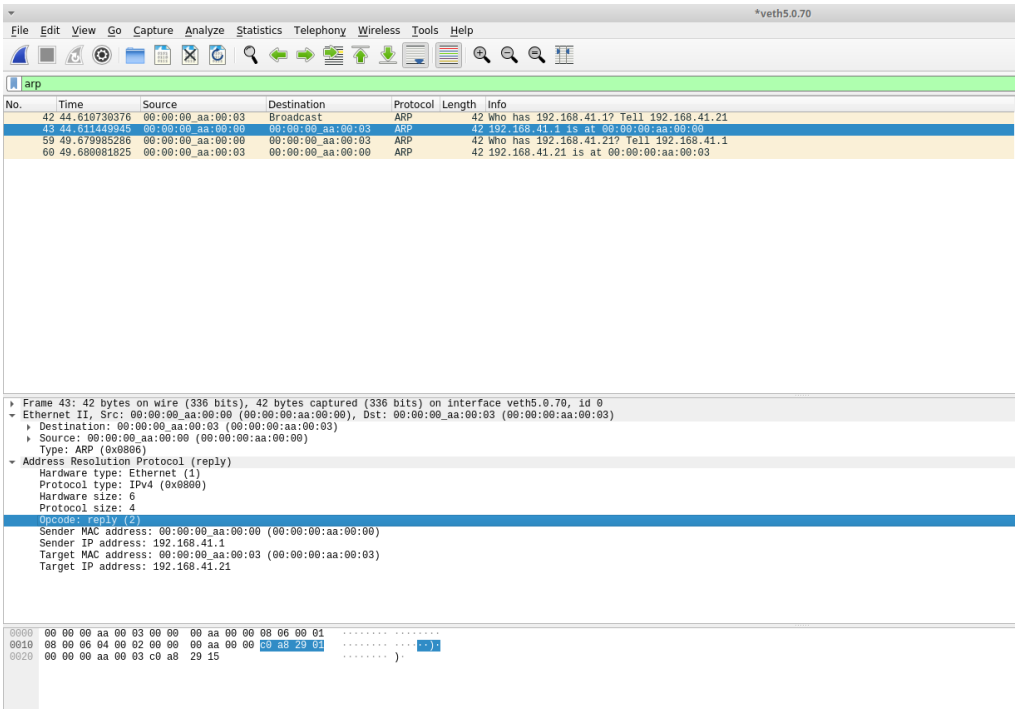

Figura 2.3.1: Captura *Wireshark* da mensagem *ARP reply*

2.3.1 Alínea a)

Qual o valor do campo ARP opcode? O que especifica?

O valor do campo *ARP opcode* é 2 que representa uma mensagem *ARP reply*.

2.3.2 Alínea b)

Em que posição da mensagem ARP está a resposta ao pedido ARP efetuado?

A resposta ao pedido *ARP* efetuado encontra-se entre os *bytes* 23 e 28, como podemos observar na figura 2.3.1

2.3.3 Alínea c)

```
root@n5:/tmp/pycore.34293/n5.conf# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.41.1             ether   00:00:00:aa:00:00   C                     eth0
root@n5:/tmp/pycore.34293/n5.conf#
```

Figura 2.3.2: Execução do comando *arp* no *host* n5

```
root@n5:/tmp/pycore.34293/n5.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.41.21  netmask 255.255.255.128  broadcast 0.0.0.0
        inet6 fe80::200:ff:feaa:3  prefixlen 64  scopeid 0x20<link>
        inet6 2001::21  prefixlen 64  scopeid 0x0<global>
        ether 00:00:00:aa:00:03  txqueuelen 1000  (Ethernet)
        RX packets 2199  bytes 180275 (180.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 26  bytes 2180 (2.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 680 (680.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 680 (680.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@n5:/tmp/pycore.34293/n5.conf#
```

Figura 2.3.3: Execução do comando *ifconfig* no *host* n5

```
root@n5:/tmp/pycore.34293/n5.conf# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.41.1    0.0.0.0         UG        0 0          0 eth0
192.168.41.0    0.0.0.0         255.255.255.128 U         0 0          0 eth0
root@n5:/tmp/pycore.34293/n5.conf#
```

Figura 2.3.4: Execução do comando *netstat -rn* no host n5

Identifique a que sistemas correspondem os endereços MAC de origem e de destino da trama em causa, recorrendo aos comandos ifconfig, netstat -rn e arp executados no PC selecionado.

Pela figura 2.3.3 verificamos que o endereço *MAC* 00:00:00:aa:00:03 corresponde ao dispositivo n5. Pelas figuras 2.3.2 e 2.3.4 sabemos que o endereço *MAC* de origem 00:00:00:aa:00:00 corresponde ao endereço *IP* 192.168.41.1 do *router* R_A.

2.3.4 Alínea d)

Justifique o modo de comunicação (unicast vs. broadcast) usado no envio da resposta ARP (ARP Reply).

O modo *unicast* envia pacotes para um dispositivo da mesma rede, ou seja, um para um, já o modo *broadcast* envia pacotes para todos os dispositivos da mesma rede, ou seja, um para todos. Foi utilizado o modo *broadcast* para saber o endereço *MAC* de um dispositivo e com isso foi enviado um pacote *ARP request* para todos os dispositivos da rede, visando obter uma resposta, *ARP reply*, do dispositivo pretendido com o seu endereço *MAC*.

2.4 Pergunta 4

Verifique se o ping feito ao segundo PC originou pacotes ARP. Justifique a situação observada

```
root@n9:/tmp/pycore.34293/n9.conf# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.169.1            ether   00:00:00:aa:00:01   C                     eth0
root@n9:/tmp/pycore.34293/n9.conf#
```

Figura 2.4.1: Execução do comando *arp* no dispositivo *n9* do departamento B

Sim, o *ping* executado pelo dispositivo *n5* do departamento A, originou pacotes *ARP* no segundo *PC*. Pela figura 2.4.1 verificamos que a tabela *ARP* do dispositivo n9, um dos dispositivos no qual o *n5* fez *ping*, possui uma entrada, confirmando que o comando ping originou pacotes *ARP*.

2.5 Pergunta 5

Identifique na mensagem ARP os campos que permitem definir o tipo e o tamanho dos endereços das camadas de rede de ligação lógica que se pretendem mapear. Justifique os valores apresentados nesses campos.

Na mensagem ARP, existem dois campos responsáveis por definir o tipo e o tamanho dos endereços das camadas de rede de ligação lógica que se pretendem mapear:

- O campo *Hardware Type* que identifica o tipo de endereço *hardware* utilizado nas camadas de rede de ligação lógica. Este possui um tamanho de 2 *bytes* e neste caso tem o valor 1, correspondente a *Ethernet*.
- O campo *Protocol Type* que define o tipo de endereço de protocolo utilizado nas camadas de rede de ligação lógica. Também possui um tamanho de 2 *bytes* e neste caso tem o valor *0x800* que representa o endereço do protolo *Ipv4*.

Relativamente, ao campo *Hardware Size* e o *Protocol Size* referem-se à quantidade de *bytes* em cada um dos tipos de endereços mencionados anteriormente: um endereço *MAC* de 6 *bytes* e um endereço *IPv4* de 4 *bytes*, como podemos observar na figura 2.3.1.

2.6 Pergunta 6

Na situação em que efetua um ping a um PC não local à sua sub-rede, esboce um diagrama em que indique claramente, e de forma cronológica, todas as mensagens ARP e ICMP trocadas, até à recepção da resposta ICMP do sistema destino (represente apenas os nós intervenientes). Assuma que todas as tabelas ARP se encontram inicialmente vazias.

Assumindo as que as Tabelas ARP se encontram inicialmente vazias, realizamos o esboço de dois diagramas nas figuras 2.6.1 e 2.6.2.

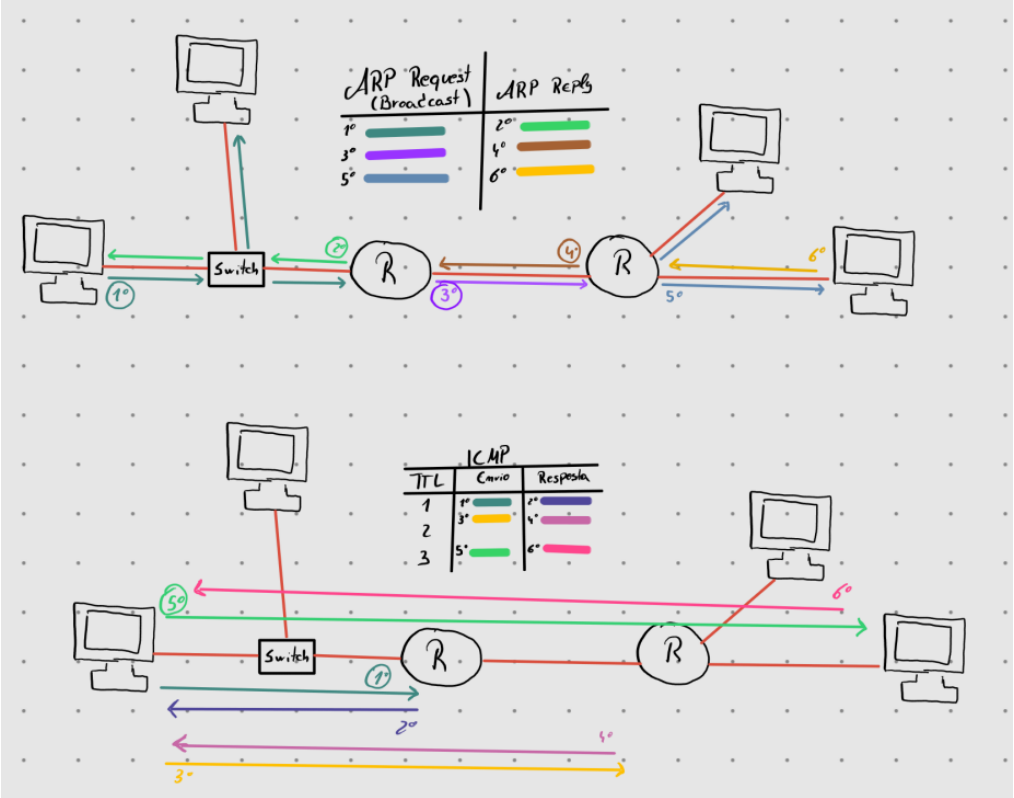

Figura 2.6.1: Diagrama mensagens ARP e ICMP trocadas entre dispositivos de sub-redes diferentes.

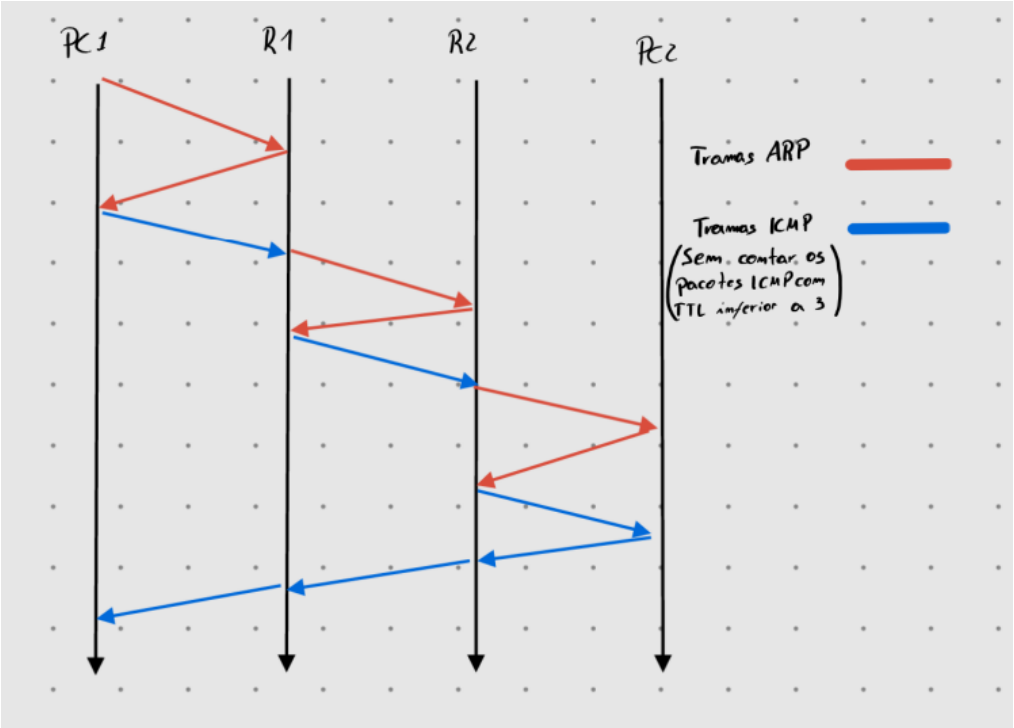

Figura 2.6.2: Diagrama cronologico de mensagens ARP e ICMP trocadas entre dispositivos de sub-redes diferentes.

Capítulo 3

Domínios de colisão

Uma rede local onde existam vários equipamentos ligados através de um meio partilhado comum constitui o que é denominado um domínio de colisão. Esta designação decorre da possibilidade de vários sistemas poderem coincidir temporalmente no envio de uma trama, causando uma interferência mútua (colisão) que deteriora as tramas originalmente enviadas.
Num domínio de colisão, apenas um dispositivo pode transmitir num determinado instante e os restantes ficam à escuta para prevenir colisões. Por esse facto, a largura de banda é partilhada entre os diversos dispositivos. Na presença de uma colisão os dispositivos envolvidos têm que retransmitir a mesma trama Ethernet algum tempo depois. As normas Ethernet implementam um método de controlo de acesso ao meio denominado CSMA/CD (estudado nas aulas teóricas), que prevê a resolução de colisões.
Os domínios de colisão existem em segmentos de rede com equipamentos interligados via hubs partilhados (repetidores) e também em redes sem fios (Wi-Fi).
As redes atuais usam maioritariamente comutadores de rede (switches) para eliminar as colisões. Conectando cada dispositivo a uma porta do comutador, cada porta constitui um domínio de colisão (se a comunicação for half-duplex) ou são eliminados se a comunicação for full-duplex. Considere a topologia de rede definida anteriormente.

3.1 Pergunta 1

Através da opção tcpdump, verifique e compare como flui o tráfego nas diversas interfaces dos vários dispositivos no departamento A (LAN comutada) e no departamento B (LAN partilhada) quando é gerado tráfego intra-departamento (por exemplo, através do comando ping). Que conclui?
Comente os resultados obtidos quanto à utilização de hubs e switches no contexto de controlar ou dividir domínios de colisão. Documente as suas observações e conclusões com base no tráfego observado/capturado.

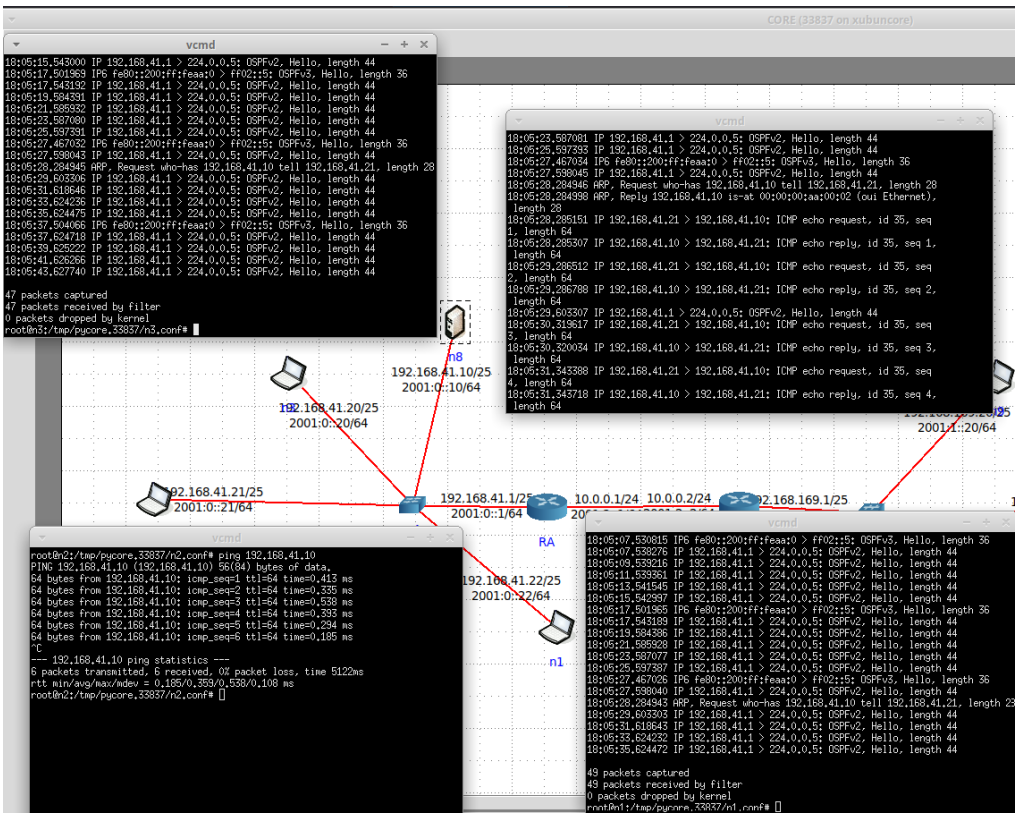

Figura 3.1.1: Execução dos comandos *tcpdump* e *ping* nos dispositivos do Departamento A

Como podemos observar na figura 3.1.1, foi executado o comando *tcpdump* nos dispositivos *n1, n3 e n8* executado o comando *Ping* do dispositivo *n2* para o servidor *n8* do Departamento A. Após análise do *output* obtido, conseguimos observar que todos os dispositivos da sub-rede receberam o *ARP request* em *broadcast* como era de esperar. Apenas o dispositivo *n2*, que realizou o comando *ping*, recebeu o *ARP Reply* e apenas o servidor *n8* recebeu os pacotes *ICMP* que lhe eram destinados.

Nesta topologia é evidente o controlo do domínio do *switch* que recebe os pacotes e reencaminha-os apenas para os hosts de destino.

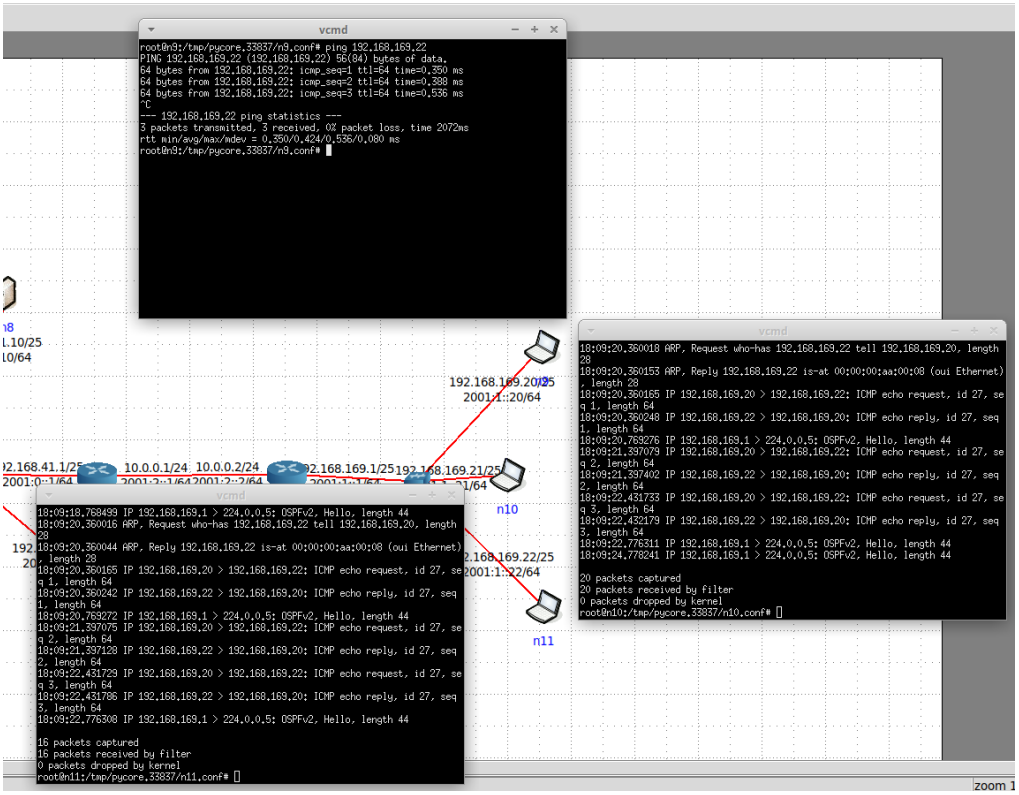

Figura 3.1.2: Execução dos comandos *tcpdump* e *ping* nos dispositivos do Departamento B

Para o Departamento B foi realizado um procedimento semelhante ao do Departamento A. Nesta topologia executamos o comando *tcpdump* nos dispositivos *n10* e *n11* e o comando *ping* no dispositivo *n9* para o dispositivo *n11*, obtendo o *output* da figura 3.1.2. Após a análise do *output* verificamos que tanto o dispositivo *n11 e n10* recebem os pacotes *ICMP*. Logo é possível evidenciar que os dispositivos *hub*, reencaminham os pacotes para todos os dispositivos da rede, não controlando os domínios de colisão de pacotes ao contrario dos *switch*.

Concluímos assim que os dispositivos *switch* é mais viável para o controlo e gestão do tráfego numa rede tendo em conta as colisões, enviando os pacotes para a *interface* onde está destinatário de um certo pacote, reduzindo assim o número de colisões.

3.2 Pergunta 2

Construa manualmente a tabela de comutação do switch do Departamento A, atribuindo números de porta à sua escolha.

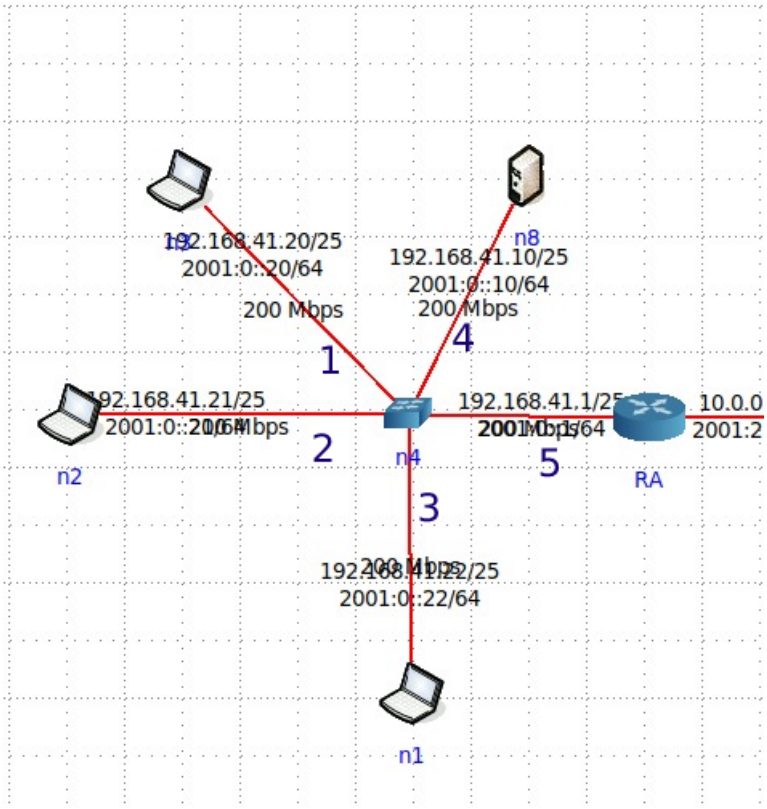

Figura 3.2.1: Identificação das interfaces relativas ao *switch*

Tabela 3.1: Tabela de comutação do *switch* com todas os dipositivos, valores de *TLL* meramente exemplificativos

Dispositivos	**Endereço MAC**	**TTL**	**Interface**
n3	00:00:00:aa:00:03	20	1
n2	00:00:00:aa:00:04	20	2
n1	00:00:00:aa:00:05	20	3
n8	00:00:00:aa:00:02	20	4
Router R_A	00:00:00:aa:00:00	20	5

Tabela 3.2: Tabela de comutação do *switch* na execução do *ping*, valores de *TLL* meramente exemplificativos

Dispositivos	**Endereço MAC**	**TTL**	**Interface**
n2	00:00:00:aa:00:04	19	2
Router R_A	00:00:00:aa:00:00	20	5

Capítulo 4

Conclusão

Na realização deste trabalho prático, tivemos oportunidade de utilizar novamente ferramentas como o *Core* e *Wireshark*. Conseguimos consolidar a matéria lecionada nas aulas teóricas sobre tramas *Ethernet*, em particular na sua análise, sobre o funcionamento do protocolo *ARP* e quais os seus casos de uso e sobre os domínios de colisão, em particular, os métodos *CSMA/CD*.

Acreditamos termos sido o mais críticos possível na realização deste trabalho, assim como termos superado todas as dificuldades encontradas nas diversas questões.